



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/769,231

01/30/2004

Cary L. Bates

ROC920030244US1

9641

46797

7590

12/08/2006

IBM CORPORATION, INTELLECTUAL PROPERTY LAW
DEPT 917, BLDG. 006-1
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER

NGUYEN, PHILLIP H

ART UNIT

PAPER NUMBER

2191

DATE MAILED: 12/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/769,231	Applicant(s) BATES ET AL.	
	Examiner Phillip H. Nguyen	Art Unit 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 30 January 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-37 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-37 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 30 January 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|--|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input checked="" type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to amendment filed on September 26, 2006. Claims 18, 23, and 24 have been amended. Claims 1-37 remain pending and have been considered below.

Specification

2. The amendment filed on September 26, 2006 overcomes the objection to paragraph [0020] of the previous office action. Therefore, the objection is withdrawn.

Claim Objections

3. The objection to claim 12 is withdrawn.

Response to Arguments

4. Applicant's arguments with respect to claims 1-37 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 18-22 and 23-37 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 18-22 are directed to a computer readable medium, which is disclosed as signal-bearing media. The specification provides intrinsic evidence the computer readable medium is intended to cover signal-bearing media (see paragraph 0022). Such is currently not believed to enable the computer readable medium to act as a computer component and realize its functionality absent being claimed in combination with the necessary hardware to receive and convert the signal-bearing media to computer readable medium.

Claims 23-37 recite a computer system, but it appears reasonable to interpret this computer system by one of ordinary skill in the art as software, per se. It is lacking of hardware components that make up a system, which enables any underlying functionality to occur other than it could be software component. Which are directed to functional descriptive material, per se, and are therefore non-statutory.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

9. Claims 17 and 36 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claims contain subject matter, which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed,

Art Unit: 2191

had possession of the claimed invention. The claims recite the limitations "*primary state...subsequent state...subsequent halted*". However, the specification does not support these limitations. The specification describes "*given state...current state...halted position*" (paragraph 0010). For examining purposes, the examiner interprets it as "*state and halted position*".

Claim Rejections - 35 USC § 102

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

11. Claims 1-21, and 23-37 are rejected under 35 U.S.C. 102(b) as being anticipated by Bates (US 7,096,458 B2).

As per claim 1:

Bates discloses a computer implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information, method comprising:

a. At a halted position in the code ("*a given debug program interrupted at a breakpoint*" Col 8, line 29. *The execution of a program is inherently halted at a breakpoint so that so that a programmer can examine the program's status, the contents of variables, and so on. Breakpoint is set and used within a debugger, and is usually implemented by inserting at that point some kind of jump, call, or*

trap instruction that transfers control to the debugger): accessing, via the debugger, a debug history repository (*"database" Col 8, line 6*) comprising a plurality of history records each containing data describing code state information for a previously encountered given state of the code (*"saves debug scenarios associated with the various source code units in a database" Col 8, line 3-5*);

b. Determining whether a current state of the code matches a previously encountered given state described in a history record in the debug history repository (*"the algorithm considers the current scenario of a breakpoint and compares it to past scenarios which have been stored" Col 7, line 60-62*);

c. Providing an indication of the match to a user (*"the match flag of the each breakpoint indicates whether the breakpoint of the previously stored scenario matches the current scenario" Col 10, line 59-61*) via the one or more interfaces whereby the user is allowed to view debug information contained in the history record corresponding to the match (*"debug session...displays information relating to the best matching scenarios" Col 8, line 10; debug session is a user interface*), the debug information describing at least an aspect of the previously encountered given state (*"a scenario may comprise: breakpoints, step locations, monitored and change variables, and set of variables including their values and locations and step locations..." Col 8, line 29-32*).

As per claim 2:

Bates discloses the method as in claim 1 above; and further discloses:

wherein the given state of the code is one at which the code was halted following execution (*"at a breakpoint, a scenario may comprise ..."* Col 8, line 29, which means the given state of code is one at which the code was halted following execution).

As per claim 3:

Bates discloses the method as in claim 1 above; and further discloses:

wherein providing the user the indicator of the match comprises displaying a selectable graphical element which (*"graphical user interface may appear during a debug session implementing features of the invention"* Co 9, line 31-32), when selected by the user through an input device (*"user then may select one of them to assist in the current debug session"* Col 8, line 15-16), reveals the debug information contained in the history record corresponding to the match (*"displays information relating to the best matching scenarios"* Col 8, line 14-15, displays information means reveals information).

As per claim 4:

Bates discloses the method as in claim 1 above; and further discloses:

the debugger and debug history repository are part of a distributed environment (*"computer 30 operates under the control of an operating system 40, and executes various computer software applications, components, programs,*

objects, modules, etc (e.g., executable program 42, calling stack 44, and debugger 50)" Col 6, line 42-45).

As per claim 5:

Bates discloses the method as in claim 1 above; and further discloses:

the code is from the group consisting of procedural program code, object oriented program code, and combinations thereof ("*executes various computer software applications, components, program, objects, modules, etc.,*" Col 6, line 42-44).

As per claim 6:

Bates discloses the method as in claim 1 above; and further discloses:

It is inherent in Bates's approach when a debugging program is executed; a breakpoint is reached which caused the code to be stopped by the debugger at the halted position.

As per claim 7:

Bates discloses the method as in claim 1 above; and further discloses:

the given state of the code is defined at least in part by an evaluation of a variable ("*expression is evaluated*" Col 9, line 23), the evaluation having been performed at the request of the user at the halted position ("*determine which ones best match at the user's request*" Col 8, line 13-14).

As per claim 8:

Bates discloses the method as in claim 7 above; and further discloses:

providing the user the indication of the match comprises displaying a series of previously performed variable evaluations performed during a previous stoppage of the code at the halted position (*"a monitored expression is a computer language expression and the location at which the expression is evaluated"* Col 9, line 21-23).

As per claim 9:

Bates discloses a computer-implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information, the method comprising:

- a. At a halted position in the code (*"a given debug program interrupted at a breakpoint"* Col 8, line 29. *The execution of a program is inherently halted at the breakpoint so that a programmer can examine the program's status, the contents of variables, and so on. Breakpoint is set and used within a debugger and is usually implemented by inserting at that point some kind of jump, call, or trap instruction that transfers control to the debugger*): accessing, via the debugger, a debug history repository (*"database"* Col 8, line 6) comprising a plurality of history records each containing
 - i) data describing code state information for a given state of the code while stopped at a halted position (*"saves debug scenarios associated with the*

- various source code units in a database" Col 8, line 3-5, a plurality of history debug scenarios are saved in a database);*
- ii) data describing a series of variable evaluations performed during a stoppage at a position in the code ("*Figure 3, item 332, and text which further expand their features Col 9, line 23-25);*
- b. performing at least one evaluation of one or more variables in the code ("*a monitored expression is a computer language expression and the location at which the expression is evaluated" Col 9, line 21-23); and*
- c. for each evaluation:
- i) determining whether a current state of the code matches a given state described in a history record in the debug history repository ("*the algorithm considers the current scenario of a breakpoint and compares it to past scenarios which have been stored" Col 7, line 60-62); and*
- ii) if so providing a user an indication of the match ("*the match flag of the each breakpoint indicates whether the breakpoint of the previously stored scenario matches the current scenario" Col 10, line 59-61) via the one or more interfaces ("debug session...displays information relating to the best matching scenarios" Col 8, line 10; debug session is a user interface).*

As per claim 10:

Bates discloses the computer-implemented method as in claim 9 above; and further discloses:

wherein providing the user the indication of the match comprises displaying a selectable graphical element which ("*debug session*" Col 8, line 16), when selected by the user through an input device, reveals the debug information contained in the history record corresponding to the match ("*displays information relating to the best matching scenarios the user then may select one of them to assist in the current debug session*" Col 8, line 14-16).

As per claim 11:

Bates discloses the computer implemented method as in claim 9 above; and further discloses:

wherein providing the user the indication of the match comprises displaying the data describing the series of variable evaluations contained in the history record corresponding to the match ("*Figure 3, item 332, and text which further expand their features* Col 9, line 23-25).

As per claim 12:

Bates discloses the computer-implemented method as in claim 9 above; and further discloses:

wherein providing the user the indication of the match comprises displaying a selectable graphical element which ("*debug session*" Col 8, line 16), when selected by the user through an input device, reveals the debug information contained in the history record corresponding to the match ("*displays information*

relating to the best matching scenarios the user then may select one of them to assist in the current debug session" Col 8, line 14-16).

As per Claim 13:

Bates discloses the method for debugging code as in claim 12 above, and further discloses:

the debug history information comprises user commentary ("displays information relating to the best matching scenarios" Col 8, line 14-15, information presented to a programmer is considered as user commentary because it gives a programmer information about the comparison between current state and previous state).

As per claim 14:

Bates discloses the method for debugging code as in claim 12 above; and further discloses:

the debug information comprises user commentary describing operations performed during a previous stoppage of the code while in a state matching the current state ("past scenarios which has been stored" Col 7, line 62, the past scenarios are user commentaries and describing the previous stoppage of code while in a state matching the current state).

As per claim 15:

Bates discloses the method as in claim 9 above; and further discloses:

after performing the at least one evaluation (*"a monitored expression is a computer language expression and the location at which the expression is evaluated"* Col 9, line 21-23), displaying a user interface configured to allow the user to create a history record for the at least one evaluation (*"if the scenarios are significantly different, the algorithm will establish and store a new scenario"* Col 3, line 19-20).

As per claim 16:

Bates discloses the method as in claim 15 above; and further discloses:

It is inherent in Bates's method because in order to create a new scenario in history record, an encountered scenario will become a given state or previous state when a new scenario is encountered in order to perform the comparison.

As per claim 17:

Bates discloses the method as in claim 15 above; and further discloses:

- i) wherein the at least one evaluation comprises a plurality of evaluations (*Figure 4, item 424 and 440, and text which further expand their features Col 9, line 48-53*); and
- ii) wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select in order to designate one of the plurality of evaluations as a primary state for the history record to be created (*for*

example Figure. 4, and text which further expand on their feature Col 9, line 48-5).

iii) wherein the state is used to determine a match between the history record to be created and a state at a halted position in the code (*"the debugger in accordance with features of the invention looks at scenarios to determine which ones best match"* Col 8, line 12-13, *a scenario is a state and used to determine the best match*).

As per claim 18:

Bates discloses a computer readable medium containing a debugger program which, when executed, performs an operation to facilitate debugging of code, the operation comprising:

at a current halted position, at which execution of the code is suspended for the purpose of allowing a user to examine the code (*"a given debug program interrupted at a breakpoint"* Col 8, line 29. *The execution of a program is inherently halted at a breakpoint so that a programmer can examine the program's status, the contents of variables, and so on. Breakpoint is set and used within a debugger and is usually implemented by inserting at that point some kind of jump, call, or trap instruction that transfers control to the debugger*):).

- a. Collecting state information related to the current stopped position (*"the algorithm considers (collecting) the current scenario of a breakpoint" Col 7, line 60-61*);
- b. although, Bates does not explicitly disclose querying a debug history repository to determine whether a current state of the code at the current halted position matches a given state described in a history record in the debug history repository. It is inherent in Bates in order to get the past scenarios to compare with the current scenario because the past scenarios is stored in database; and
- c. If so, providing a user an indication of the match via the one or more interfaces (*"the match flag of the each breakpoint indicates whether the breakpoint of the previously stored scenario matches the current scenario" Col 10, line 59-61*).

As per claim 19:

Bates disclose the computer readable medium as in claim 18 above; and further discloses:

wherein the debugger program allows a user to enter a new record in the debug history repository if a matching record is not found (*"in accordance with features of the invention, as will be described, greater freedom and flexibility is granted to the computer programmer so she/he can create...debug scenarios..." Col 7, line 53-55*).

Art Unit: 2191

As per claim 20:

Bates discloses the computer readable medium as in claim 18 above; and further discloses:

wherein the debugger program automatically creates a new record in the debug history repository if a match record is not found (*"if the scenarios are significant different, the algorithm will establish and store a new scenario" Col 3, line 19-20*).

As per claim 21:

Bates discloses the computer readable medium as in claim 18 above; and further discloses:

the state information is defined at the least in part by the evaluation of one or more variables (*"expression is evaluated" Col 9, line 23*), the evaluation having been performed at the request of the user at the halted position (*"determine which ones best match at the user's request" Col 8, line 13-14*)..

As per claim 23:

Bates discloses a computer system for debugging code comprising:

- a. a debug history repository (*"database" Col 8, line 6*) comprising a plurality of history records each containing data describing code state information for a given state of the code under debug (*"saves debug scenarios associated with the various source code units in a database" Col 8, line 3-5*);

b. A debugger comprising one or more interfaces for specifying debugger operations and displaying debug information (*Figure 4, and text which further expand their features Col 9, line 32-35;*

wherein the debugger is configured to:

- i) create the plurality of history records in the debug history repository (*"saves debug scenarios associated with the various source code units in a database" Col 8, line 3-5;*
- ii) *it is inherent in Bates's invention because in order to perform a comparison between the current scenario and the past scenarios, a query must be performed on a database to access the past scenarios;*
- iii) determine, at a given halted position in the code (*"a given debug program interrupted at a breakpoint" Col 8, line 29*), whether a current state of the code matches a given state described in a history record of the plurality of history records (*"the algorithm considers the current scenario of a breakpoint and compares it to past scenarios which have been stored" Col 7, line 60-62*); and
- iiii) if a matching state is determined, provide a user an indication of the match (*"the matched flag of the each previously stored scenario matches the current scenario" Col 10, line 59-61*) via the one or more interfaces while at the given halted position (*Figure 4, and text which further expands their features Col 9, line 31-35*) whereby the user is allowed to view debug information contained in the history record corresponding to

the match ("*displays information relating to the best matching scenarios*"
Col 8, line 14-15).

As per claim 24:

Bates discloses the computer readable medium as in claim 23 above; and further discloses:

wherein the repository is a database selected from the group consisting of a relational database, object-relational database, extensible-markup-language (XML) database, and combination thereof ("*database*" *Col 8, line 5*).

As per claim 25:

Although, Bates discloses the computer readable medium as in claim 23 above, but does not explicitly disclose:

wherein the debugger queries the debug history repository based on available current state information. It is inherent in Bates's approach. In order to query the database, the current scenario information must be used for comparing with the past scenarios.

As per claim 26:

Bates discloses the computer system as in claim 23 above; and further discloses:

herein the one or more interfaces comprise a graphical user interface ("*user interface is a graphical one*" *Col 9, line 34*).

As per claim 27:

Bates discloses the computer system as in claim 23 above; and further discloses:

wherein providing the user the indicator of the match comprises displaying a selectable graphical element which (*Figure 4, and text which further expands their features Col 9, line 31-54*), when selected by the user through an input device, reveals the debug information contained in the history record corresponding to the match (*Figure 4, and text which further expands their features Col 9, line 31-55*).

As per claim 28:

Bates discloses the computer system as in claim 27 above; and further discloses:

the debug history information comprises user commentary ("*displays information relating to the best matching scenarios*" Col 8, line 14-15, *information presented to a programmer is considered as user commentary because it gives a programmer information about the comparison between current state and previous state*).

As per claim 29:

Bates discloses the computer system as in claim 27 above; and further discloses:

the debug information comprises user commentary describing operations performed during a previous stoppage of the code while in a state matching the current state ("*past scenarios which has been stored*" Col 7, line 62, *the past*

scenarios are user commentaries and describing the previous stoppage of code while in a state matching the current state).

As per claim 30:

Bates discloses the computer system as in claim 23 above; and further discloses:

each history record further contains data describing a series of variable evaluations performed during a stoppage at a position in the code (*"Figure 3, item 332, and text which further expand their features Col 9, line 23-25).*

As per claim 31:

Bates discloses the computer system as in claim 30 above; and further discloses:

- i) wherein the debugger is further configured to perform at least one evaluation of one or more variables in the code at the halted position (*"monitored expression is a computer language expression and the location at which the expression is evaluated" Col 9, line 21-22);*
- ii) wherein the debugger performs the determination of whether the current state of the code matches a given state described in a history record for each evaluation (*"determine if the current debug scenario is a commonplace occurrence or is a new phenomena" Col 8, line 8-10);* and
- iii) wherein, if the matching state is determined and the indication of the match provided (*"the matched flag of the each breakpoint indicates whether the breakpoint of the previously stored scenario matches the current scenario" Col*

Art Unit: 2191

10, line 59-61), the user is allowed to view the data contained in the history record corresponding to the match (*Figure 4, and text which further expands their features Col 9, line 31-55*) and describing the series of variable evaluations performed during the stoppage at a position in the code (*Figure 4, item 424, 440*).

As per claim 32:

Bates discloses the computer system as in claim 30 above; and further discloses:

the debug history information comprises user commentary ("*displays information relating to the best matching scenarios*" Col 8, line 14-15, *information presented to a programmer is considered as user commentary because it gives a programmer information about the comparison between current state and previous state*).

As per claim 33:

Bates discloses the computer system as in claim 30 above; and further discloses:

the debug information comprises user commentary describing operations performed during a previous stoppage of the code while in a state matching the current state ("*past scenarios which has been stored*" Col 7, line 62, *the past scenarios are user commentaries and describing the previous stoppage of code while in a state matching the current state*).

Art Unit: 2191

As per claim 34:

Bates disclose the computer system as in claim 30 above; and further discloses:

the debugger is configured to display a user interface configured to allow the user to create a history record for the at least one evaluation (*"if the scenarios are significantly different, the algorithm will establish and store a new scenario"* Col 3, line 19-20).

As per claim 35:

Is a computer system claim reciting the same limitation as in claim 16, and therefore, has been addressed in connection with the rejection of claim 16.

As per claim 36:

Bates discloses the computer system as in claim 34 above; and further discloses:

- i) wherein the at least one evaluation comprises a plurality of evaluations(*Figure 4, item 424 and 440*);
- ii) wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select in order to designate one of the plurality of evaluations as a primary state for the history record to be created (*Figure 4*); and
- iii) wherein the state is used to determine a match between a state described by code state information contained in the history record to be created and a subsequent state at a subsequent halted position in the code (*"the algorithm*

considers the current scenarios of a breakpoint and compares it to past scenarios which have been stored" Co7, line 60-62).

As per claim 37:

Bates discloses the computer system as in claim 34 above; and further discloses:

- i) wherein the at least one evaluation comprises a plurality of evaluations ("*Figure 4, item 424 and 440*"); and
- ii) wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select in order to designate which of the plurality of evaluations are to be included in the history record to be created (*Fig. 4, item 424, 440*).

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

13. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bates et al. (US 7,096,458 B2) in view of Novik et al (US 6,367,034).

As per claim 22:

Bates discloses the computer readable medium as in claim 18 above.

Bates does not explicitly disclose:

wherein the debug history repository is queried using filters selected from the group consisting of developer name, time, and combinations thereof.

However, Novik discloses an analogous method that performs:

the debug history repository is queried using filters selected from the group consisting of developer name, time, and combinations thereof (*"a conventional query language, such as that typically used to retrieve data associated with relational databases, can be extended to filter events on the basis of the time of their occurrence" Col 3, line 42-45*).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to incorporate the teachings of Novik into Bates's approach. One of ordinary skill in the art would have been motivated to include time intervals in the query because this feature is particularly useful to manage the reporting of frequent events that would otherwise consume large amounts of resources if the events were to be individually reported (see Novik Col 3, line 47-50).

CONCLUSION

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Bates et al (6,077,312) discloses apparatus, program product and method of debugging utilizing a context sensitive breakpoint.

b. Bates et al (US 6,854,073 B2) discloses debugger program time monitor.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Friday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
11/9/06

Wei Zhen
Supervisory Patent Examiner

Mary Stuelmen
Primary Examiner
11-27-2006